Assignment 1

1 Rules of the Game

- You work on this assignment alone, no groups of students are allowed.
- Your solution will be evaluated with points ranging from 0 to 15.
- You have to upload your solution to this assignment by 20.10.2013. After this date, **you lose 3 points for each started week of delay**. In exceptional and justified cases (e.g. long-term disease) we decide how to proceed on individual basis. In that case, write me an email at petr.kremen@fel.cvut.cz.
- The solution of the assignment has to be uploaded through the **web application** `http://cw.felk.cvut.cz/upload`. Please, upload the ZIP archive containing:
  - one file `.pdf` – answers to the questions in the Section 2.1
  - one or more file(s) `.owl` – final ontology developed by you in Section 2.2.
  - more file(s) `.rq` – SPARQL queries developed by you in Section 2.3.
- To complete the assignment, you should be familiar with the Web Ontology Language to the extent of the OWL 2 primer (`http://www.w3.org/TR/owl2-primer`) and with the SPARQL 1.1 language

2 Assignment

2.1 Analysis

2.1.1 General Questions

Consider an ontology (description logic theory) $\mathcal{K} = \mathcal{T} \cup \mathcal{A}$ with a TBox containing three axioms:

$$\begin{aligned} \mathcal{T} \quad = \quad & \{Man \sqcup Woman \sqsubseteq Person, \\ & Man \sqsubseteq \neg Woman \sqcap \exists hasWife \cdot Woman\}, \end{aligned}$$

and $\mathcal{A} = \emptyset$. For each question, you can use Protégé to verify your findings.

1. Construct any model $\mathcal{I} = (\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}})$ of $\mathcal{K}$ for which $|\Delta^{\mathcal{I}}| = 3$ and $|Woman^{\mathcal{I}}| = 2$. If it is not possible, explain why in detail.

2. Reduce the subsumption problem $\mathcal{K} \models (Man \equiv \exists hasWife \cdot \top)$ into the consistency checking problem, i.e. construct an ontology, inconsistency of which indicates that the subsumption holds.

3. Both the "tree model property" (at least one model of the concept is tree-shaped) and the "finite model property" (at least one model of the concept is finite) allows tableau algorithm implementations using efficient optimizations and thus speed-up the reasoning process. These two properties do not hold for all description logics.

   Find out for each of the three theories:

$$\begin{aligned} &\mathcal{K} \\ &\mathcal{K}_1 &= \mathcal{K} \cup \{Woman \sqsubseteq \left(\leq 1\, hasWife^{-}\right)\} \\ &\mathcal{K}_2 &= \mathcal{K}_1 \cup \{hasWife \sqsubseteq hasSpouse\} \end{aligned}$$

   whether it employs the tree model property and whether it employs the finite model property. You can use the Description Logic Complexity Navigator at `http://www.cs.man.ac.uk/~ezolin/dl` to answer this question.

2.1.2 Analysis of Existing Ontology

Explore the ontologies at the following URLs:

- `http://krizik.felk.cvut.cz/ontologies/2011/general-family.owl`.
- `http://krizik.felk.cvut.cz/ontologies/2011/father-without-children.owl`.

We will abuse the terminology and use description logic terms and OWL terms interchangably (see `https://cw.felk.cvut.cz/wiki/_media/courses/ae4m33rzn/protege-crash-course.pdf` for more details). Please use the description logic notation when solving the following tasks.

1. Ontology `father-without-children.owl` is of expressiveness $\mathcal{ALC}$. Check consistency of the ontology by means of the tableau algorithm. Describe in detail and visualize the run of the algorithm.

2. Using the tableau algorithm and some of the error-explanation algorithm according to your choice (Reiter algorithm, CS-tree algorithm) find all minimal unsatisfiability preserving sets for the unsatisfiable class $FatherWithoutChildren$. Describe in detail and visualize the run of the error-explanation algorithm (You do not need to visualize show the tableau algorithm any more). Check the correctness of your results using the OWL reasoner Pellet.

3. Explain, why $PETR$ is not an (inferred) instance of the class $ParentOfAtLeastTwoChildren$, although it occurs in **two** axioms of the form $hasChild(PETR, \bullet)$, i.e. $hasChild(PETR, OLGA)$ and $hasChild(PETR, JIRI)$. Find at least two ways how to adjust the ontology so that $PETR$ becomes an instance of $ParentOfAtLeastTwoChildren$.

2.2 Synthesis of Own Ontology – Genealogical Tree of a Well-Known (e.g. Aristocratic) Family

When implementing tasks in this part, use the ontology `http://krizik.felk.cvut.cz/ontologies/2011/general-family.owl`, as a starting point (do not use `owl:imports`). Rename the ontology to the $< FEL\!-\!username > -family.owl$ The resulting ontology must be consistent.

1. Using Protégé refactoring move all axioms, except those causing unsatisfiability of $FatherWithoutChildren$ class, from the imported `father-without-children.owl` ontology into your ontology.

2. Specify characteristics (reflexivity, asymmetry, etc.) and define inverses of the object properties $hasChild$ and $hasSibling$.

3. Formalize the object properties $hasDescendant$ and $hasAncestor$ that will be used for inferring descendants/ancestors into arbitrary depth. E.g. it will be possible to infer $hasAncestor(JIRI, MIRKO)$.

4. Define the class of "all parents, that have at least 3 children, but at most 1 daughter that has exactly two grandsons or one granddaughter.".

5. Finalize the ontology for a genealogical information system – add and axiomatize at least 5 more classes and 3 more properties (both object and data properties are required). Several suggestions:

   a) family relationships – relationships of being spouse, relationship of being an uncle of someone, relationships of being brother-in-law.

   b) genealogical data – date of birth, place of birth.

   Define classes and relationships in such a way that you can easily use them for query formulation in section 2.3.

6. Develop a genealogical tree (at least 3 generations) of a known historical family (see e.g. `http://www.burkespeerage.com/articles/scotland/page31d.aspx`) and check adequacy of the ontology you developed in the previous point.

2.3 Querying the Ontology

For each query you developed in this part (i) write its SPARQL form into a separate .rq file, (ii) test on the developed ontology using the Pellet inference engine of version 2.3 (http://pellet.owldl.com), (iii) write its results into a comment (#) of the .rq file. Next identify queries that can be answered by using the DL query tab in Protégé, and those for which full conjunctive query engine is necessary.

1. Create a query that finds all pairs of persons being in brother-in-law/sister-in-low relationship, and, at the same time, each having at least one sibling.

2. Create a query that finds out whether there exists (or can be inferred) at least one person, at least one daughter of which has a son. We are interested just in the existence, not in their identity.

3. Show, how the previous query could be evaluated only by means of the standard tableau algorithm for consistency checking, if there is not inference engine for conjunctive queries available (e.g. in the DL Query tab in Protégé).